CS 595: Assignment #3

Due on Thursday, October 2, 2014

Dr Nelson 4:20PM

VICTOR NWALA

Contents

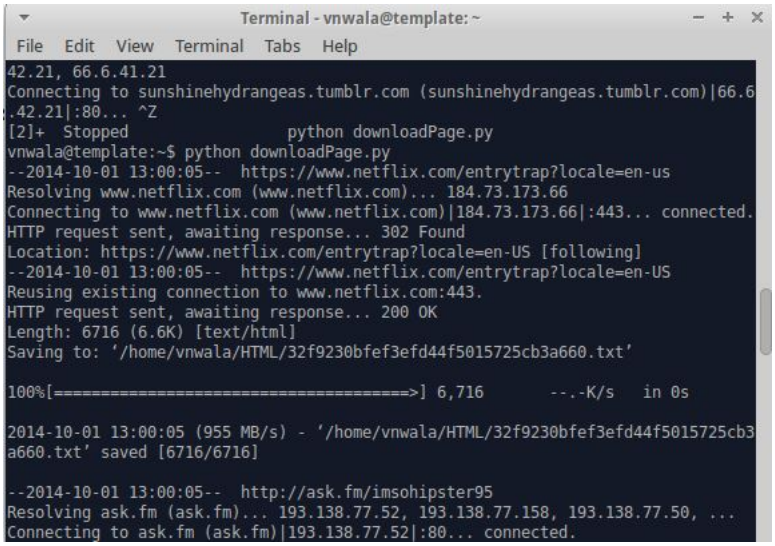

Figure 1: downloadPage.py at work

Problem 1

1. Download the 1000 URIs from assignment #2. "curl", "wget", or "lynx" are all good candidate programs to use. We want just the raw HTML, not the images, stylesheets, etc.

Listing 1: Python script to download html pages

```
import hashlib
from hashlib import md5
import os

fh = open("UniqueURLS.txt",'r')

for line in fh:
     url=line
     url=url.replace('\n','')
10
     def computeMD5hash(message):
          m = hashlib.md5()
15        m.update(message)
          return m.hexdigest()

20   hashMessage = computeMD5hash(url)
     print hashMessage

     os.system("wget -O /home/vnwala/Html/" + hashMessage + ".txt "+ url)
```

This script downloads html pages, saves them in text files and places them in a folder called Html. The files are also saved with their respective md5 encrypted names.

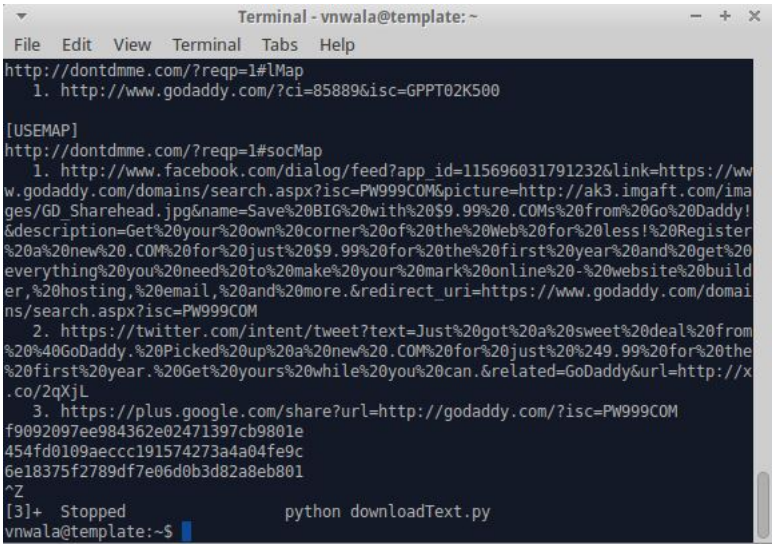

Figure 2: downloadText.py at work

Listing 2: Python script to get text from html pages

```
import hashlib
from hashlib import md5
import os

fh = open("UniqueURLS.txt",'r')

for line in fh:
     url=line
     url=url.replace('\n','')




     def computeMD5hash(message):
          m = hashlib.md5()
          m.update(message)
           return m.hexdigest()


     hashMessage = computeMD5hash(url)
     print hashMessage

     os.system("lynx -dump -force_html " + url+ "  > /home/vnwala/TextFiles/" +
         hashMessage  +".processed"+ ".txt ")
```

This script downloads the text content of all the urls and stores the with the md5 names into a file.

Listing 3: Python script store URI and their md5 names

```
import hashlib
from hashlib import md5
import os
```

Problem 1 continued on next page...

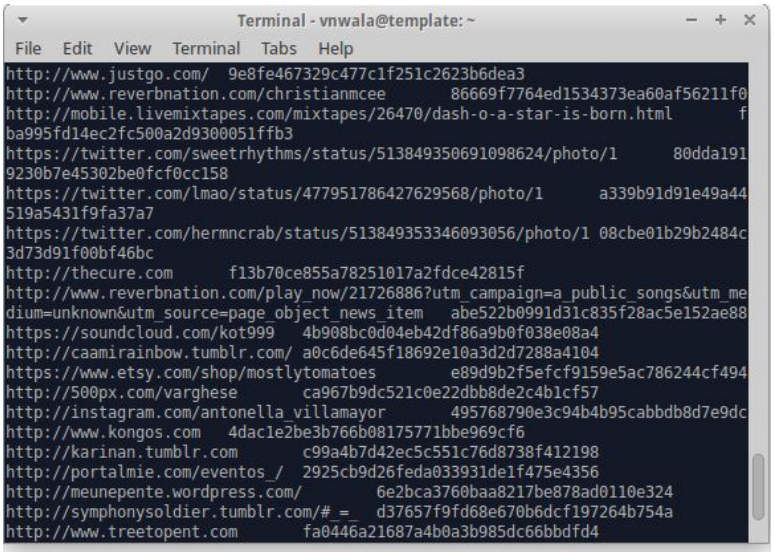

Figure 3: getHashes.py at work

```
fh = open("UniqueURLS.txt",'r')

for line in fh:
     url=line
     url=url.replace('\n','')




     def computeMD5hash(message):
          m = hashlib.md5()
          m.update(message)
           return m.hexdigest()



     hashMessage = computeMD5hash(url)

     i = url +"\t"+hashMessage
        print i
     saveFile = open("hashes.txt",'a')
     saveFile.write(i)
     saveFile.write('\n')
     saveFile.close()
```

This script converts URIs to md5 encryption and stores the mapping.

Problem 2

2. Choose a query term (e.g., "shadow") that is not a stop word (see week 4 slides) and not HTML markup from step 1 (e.g., "http") that matches at least 10 documents (hint: use "grep" on the processed files). If the term is present in more than 10 documents, choose any 10 from your list. (If you do not end up with a list of 10 URIs, you've done something wrong).

Problem 2 continued on next page...

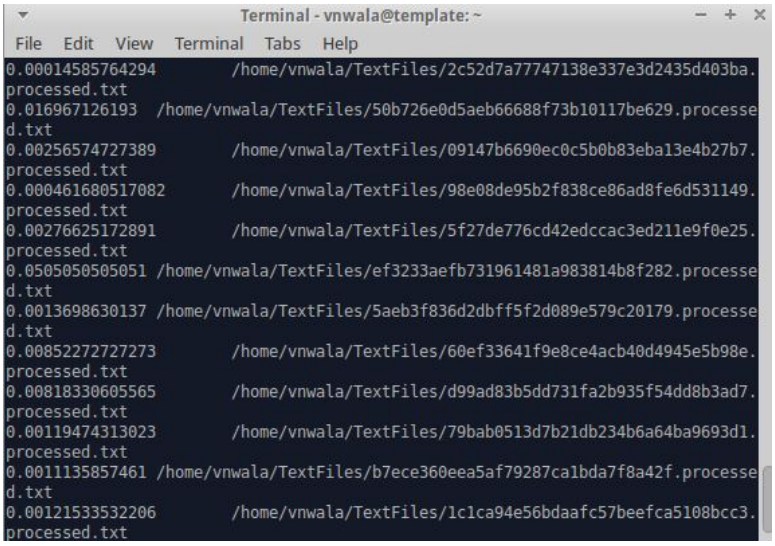

Figure 4: wordCount.py at work

Listing 4: Python script to calculate TF (wordCount.py)

```
import os
import glob
import subprocess
import re
from decimal import *

fh = glob.glob("/home/vnwala/TextFiles/*.txt")
for line in fh:
     url=line
     url=url.replace('\n','')
     proc = subprocess.Popen(["wc -w " + url], stdout=subprocess.PIPE, shell=True)
        (out, err) = proc.communicate()
     index = out.find(" ")
     index = out[:index]
     proc = subprocess.Popen(["grep -c 'football' " + url], stdout=subprocess.PIPE,
         shell=True)
        (out, err) = proc.communicate()
     number = out
     number=number.replace('\n','')
     if (number > "0" and index > "0"):

          TF =(float(number)/int(index))
          k = str(TF) + '\t' + url
          print k
          saveFile = open('tpCal.txt','a')
          saveFile.write(k)
          saveFile.write('\n')
          saveFile.close()
```

This script get the total word count of my text file, queries the word “football” in all the text files, it then calculates TF for each file by diving the frequecy of the word (football) by the total word count for each file and stores the result.

Problem 2 continued on next page. . .

To calculte IDF, I used searched for the word "football", I got 381,000,000 results. Still using the assumption that 20 billion pages are indexed by Bing:
IDF = logarithm to the base of 2 0f the result of (20,000,000,000/381,000,000)
IDF = 5.71407
I chose 10 URI randomly and ranked them. The table displays ranking from the highest to the lowest.

Table 1: URL RANKING

TF-IDF	TF	IDF	URI
0.078060999	0.0136612021858	5.71406519205585	http://npcironmen.com
0.046759947	0.00818330605565	5.71406519205585	http://www.maxpreps.com/national/national.htm
0.044381088	0.00776699029126	5.71406519205585	http://qoly.jp
0.028244428	0.00494296577947	5.71406519205585	http://www.chelseafc.com
0.006363106	0.0011135857461	5.71406519205585	http://www.eonline.com
0.005771783	0.0010101010101	5.71406519205585	http://twinsdaily.com/
0.0029198084	0.00051098620337	5.71406519205585	http://www.cracked.com
0.002835764	0.00049627791563	5.71406519205585	http://www.latimes.com/entertainment/
0.0026380726	0.00046168051708	5.71406519205585	http://vkdiz.ru
0.0008334401	0.00014585764294	5.71406519205585	http://www.lazywrita.com

Problem 3

3. Now rank the same 10 URIs from question #2, but this time by their PageRank. Use any of the free PR estimaters on the web, such as:
http://www.prchecker.info/check_page_rank.php
http://www.seocentro.com/tools/search-engines/pagerank.html
http://www.checkpagerank.net/
To answer this question, I used http://www.prchecker.info/check_page_rank.php, to check the ranks of the respective URIs. The table displays ranking from the highest to the lowest.

Table 2: URL RANKING USING http://www.prchecker.info/check_page_rank.php

URI	PAGE RANK	NORMALIZED VALUES
http://www.maxpreps.com/national/national.htm	8/10	1
http://www.chelseafc.com/	7/10	0.875
http://www.latimes.com/entertainment/	7/10	0.875
http://www.eonline.com	7/10	0.875
http://www.cracked.com	6/10	0.75
http://qoly.jp	5/10	0.675
http://twinsdaily.com/	4/10	0.5
http://vkdiz.ru	2/10	0.25
http://npcironmen.com	0/10	0
http://www.lazywrita.com	0/10	0

Comparing both ranking schemes in 2 and 3, of the URIs I noticed they were not the same but in some way consistent except few exceptions. Some links rank high on both methods, some intermediate on both and others low, even if their ranking positions differ. I observed that links which have not been indexed by google

Problem 3 continued on next page...

do not have a page rank using the PageRank scheme. Also the ranking system of this scheme is based on counting the number and quality of links to a page, hence giving them a rank in their order of importance.

Problem 3 continued on next page...

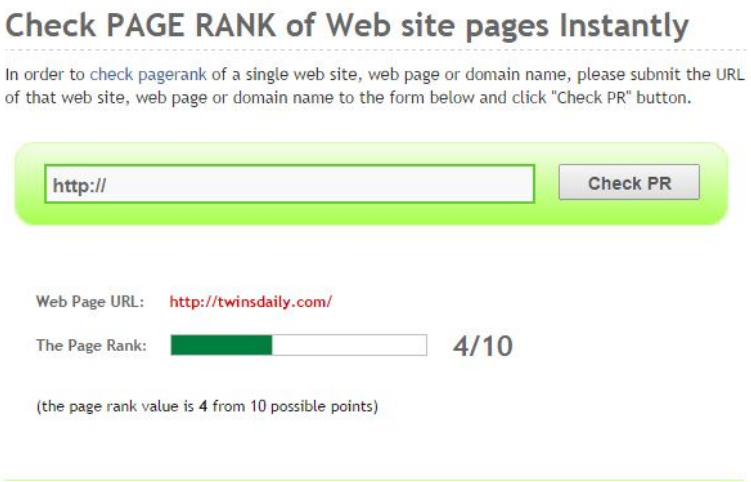

Figure 5: PAGE RANK SCREEN SHOT